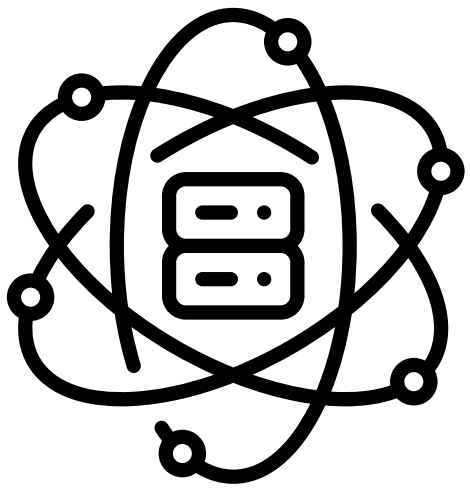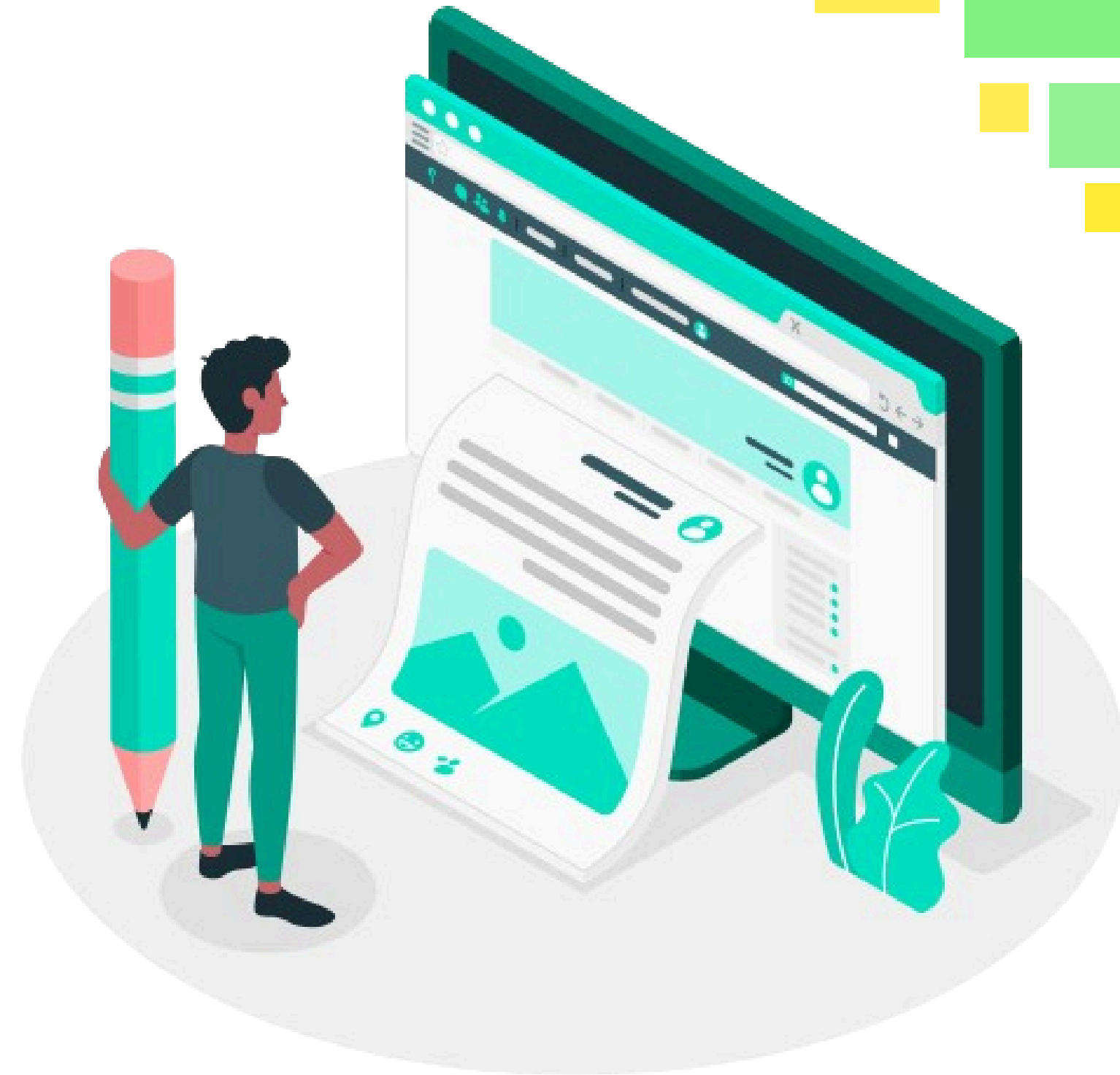# Codveda

Data Science

Content

1. **About Us**
2. **Instructions**
3. **Submission**
4. **Task List**

About Us

Welcome to Codveda Technology, where innovation meets excellence. Founded with a vision to empower businesses through cutting-edge IT solutions, we specialize in delivering tailored services that drive success in the digital era.

At Codveda, we offer a diverse range of services, including web development, app development, digital marketing, SEO optimization, AI/ML automation, and data analysis.

Our team of skilled professionals is committed to helping businesses unlock their full potential by providing innovative, scalable, and reliable solutions.

INSTRUCTIONS

- **Update your LinkedIn profile with your achievements, including the offer letter and completion certificate. Mention and tag @Codveda in your posts.**
- **Use hashtags like #CodvedaJourney, #CodvedaExperience, and #FutureWithCodveda to showcase your progress and experiences.**
- **Share your project completion updates on LinkedIn, accompanied by a video explanation and the GitHub project repository link.**
- **You will be provided with four tasks. Select and complete any three tasks within your domain to fulfill the internship requirements.**
- **Submit your completed tasks via the Codveda submission form. Ensure all tasks are submitted within the allocated 15-day period.**

SUBMISSION

- Create a professional video showcasing your internship projects and achievements.
- Host the video on LinkedIn to provide proof of your work and establish credibility among your peers. Consider tagging Codveda Technology in your posts to ensure they are notified of your work using hashtags like #CodvedaAchievements and #CodvedaProjects.
- A SUBMISSION FORM will be shared later. Till then, please continue your tasks and maintain a separate file for each level.
- When posting the video on LinkedIn, include engaging content that highlights your contributions and skills. Tailor the post to your specific internship domain to maximize impact and visibility.

Level 1 (Basic)

Task 1: Data Collection and Web Scraping

- Description: Collect data from a website using web scraping techniques.

Objectives:

- Identify a target website and inspect its structure.
- Use BeautifulSoup and requests libraries to scrape data from web pages.
- Store the scraped data in a structured format (e.g., CSV, JSON).
- Handle common challenges such as pagination and dynamic content.
- Tools: Python, BeautifulSoup, requests, pandas.

Level 1 (Basic)

Task 2: Data Cleaning and Preprocessing

- Description: Clean and preprocess a raw dataset to make it suitable for analysis.

Objectives:

- Handle missing data (e.g., imputation, removal).
- Detect and remove outliers.
- Convert categorical variables into numerical format using one-hot encoding or label encoding.
- Normalize or standardize numerical data.
- Tools: Python, pandas, scikit-learn

Level 1 (Basic)
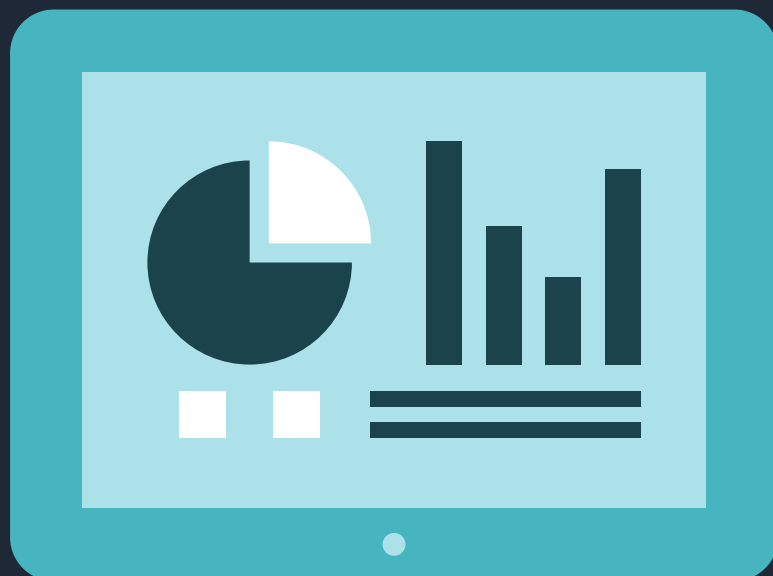
Task 3: Exploratory Data Analysis (EDA)

- Description: Perform exploratory data analysis to understand the underlying structure and trends in the data.

Objectives:

- Compute summary statistics (mean, median, variance, etc.).
- Visualize the data using histograms, scatter plots, and box plots.
- Identify correlations between numerical features using a correlation matrix.
- Generate a report summarizing insights from the EDA.
- Tools: Python, pandas, matplotlib, seaborn.

Level 2 (Intermediate)

Task 1: Predictive Modeling (Regression)

- Description: Build and evaluate a regression model to predict a continuous variable (e.g., house prices).

Objectives:

- Split the dataset into training and testing sets.
- Train a linear regression model using scikit-learn.
- Evaluate the model using performance metrics like mean squared error (MSE) and R-squared.
- Experiment with multiple models (e.g., Decision Trees, Random Forest) and compare performance.
- Tools: Python, scikit-learn, pandas, matplotlib.

Level 2 (Intermediate)

Task 2: Classification with Logistic Regression

- Description: Build a decision tree classifier to predict a categorical outcome (e.g., predict species of flowers).

Objectives:

- Preprocess the data (e.g., handling categorical features, feature scaling).
- Train and evaluate the logistic regression model.
- Use metrics such as accuracy, precision, recall, and the ROC curve for evaluation.
- Compare logistic regression with other classifiers like Random Forest or SVM.
- Tools: Python, scikit-learn, pandas, matplotlib.

Level 2 (Intermediate)
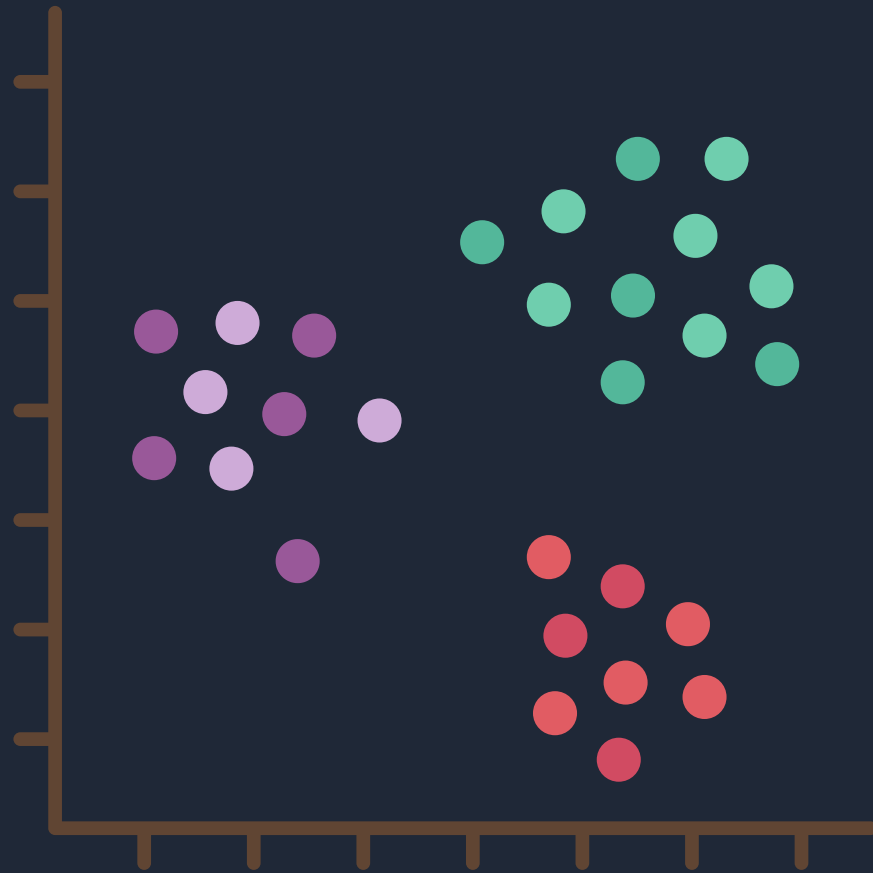
Task 3: Clustering (Unsupervised Learning)

- Description: Implement K-Means clustering to group data points into clusters without labels (e.g., customer segmentation).

Objectives:

- Apply K-Means clustering to the dataset.
- Use the elbow method or silhouette score to determine the optimal number of clusters.
- Visualize the clusters in 2D space using PCA or t-SNE for dimensionality reduction.
- Interpret the clustering results and summarize key findings.
- Tools: Python, scikit-learn, matplotlib, seaborn.

Level 3 (Advanced)

Task 1: Time Series Analysis

- Description: Analyze and model time-series data to forecast future values (e.g., stock prices, sales).

Objectives:

- Plot and decompose the time series into trend, seasonality, and residual components.
- Implement moving average and exponential smoothing techniques.
- Build an ARIMA or SARIMA model for forecasting.
- Evaluate the model using metrics such as RMSE and visualize the forecast.
- Tools: Python, pandas, statsmodels, matplotlib.

Level 3 (Advanced)

Task 2: Natural Language Processing (NLP) - Text Classification

- Description: Classify text data into categories (e.g., spam vs. non-spam, sentiment analysis).

Objectives:

- Preprocess text data (tokenization, removing stopwords, stemming/lemmatization).
- Convert text into numerical representation using TF-IDF or Word2Vec.
- Train a classification model (e.g., Naive Bayes, Logistic Regression) on the processed text.
- Evaluate the model using precision, recall, and F1-score.
- Tools: Python, nltk, scikit-learn, pandas.

Level 3 (Advanced)

Task 3: Neural Networks with TensorFlow/Keras

- Description: Build and train a simple feed-forward neural network to classify images or structured data.

Objectives:

- Load a dataset (e.g., MNIST digits or a structured dataset) and preprocess it.
- Design a neural network architecture using TensorFlow or Keras.
- Train the model using backpropagation and evaluate it using accuracy and loss curves.
- Tune hyperparameters (e.g., learning rate, batch size) to improve performance.
- Tools: Python, TensorFlow, Keras, pandas, matplotlib.

How to Contact Us?

For additional information, kindly get in touch with our team.

in   @codveda

✉   support@technofyz.com

🌐   www.codveda.com